

REMARKS

Claims 1-32 are pending and are unamended.

Entry of Rule 116 Response

Entry of this response is requested because this response does not raise any new issues that would require further consideration and/or search. No new claims are being presented in this response. No new matter is raised by this response. This response could not have been previously presented because the outstanding § 112 and § 103(a) rejections are based on new reasoning. Lastly, it is requested that the response be entered even if the application is not allowed because this response will place the application in better form for appeal by materially simplifying the issues.

If the application is not in proper form for allowance, Applicants request that the Examiner telephone the undersigned to discuss any further outstanding issues.

Request for Interview Prior to Formal Action on Amendment

Applicants request an interview prior to formal action on this response. An “Applicant Initiated Interview Request Form” accompanies this response. Please contact Applicants’ undersigned representative to schedule the interview.

Claim Objections

The Examiner objects to claims 1-15 as being substantially duplicates of claims 16-28. Applicants respectfully traverse this objection. In claim 1, each node includes “one or more processors.” In claim 16, each node includes (i) a processor subsystem including at least one processor, and (ii) an operating system. Accordingly, an “operating system” is not a required limitation of claim 1. Although a processor or processor subsystem typically uses an operating system to function, a processor and an operating system are still physically separate elements. See, for example, Figure 5e. Claims 1 and 16 are not merely claiming the same invention twice,

using synonymous wording. Applicants have chosen to claim two different embodiments of the present invention, one with an operating system, and one without an operating system.

Furthermore, an operating system is not an essential element of the present invention. The processor could be hardwired or could use firmware to perform the functions that would otherwise be performed by the operating system. Applicants do not wish to restrict claim 1 to only to schemes that use an operating system, when other schemes may be employed in place of the operating system.

35 U.S.C. § 112, second paragraph, rejection

Claims 1-32 were rejected because it is unclear as to how all of the processors can be located at a single node. Applicants traverse this rejection.

It is very common to build processing systems where all of the processors are located at a single node¹. The left-hand portions of Figures 5a-5e each illustrate unsplit systems where all of the processors are located at a single node. See the description of Figures 5a-5e on page 6, lines 10-24 of the specification, and in the subsequent detailed description of these figures. There is inherently not a plurality of nodes in an unsplit system. Stated simply, an unsplit system always has only one node regardless of how many processors exist at the one node. In contrast, a split system inherently always has a plurality of nodes. The remaining portions of Figure 5a-5e show split systems. In a split system, each node can have one or more processors.

Referring to the Examiner's inquiry, all of the processors are located at a single node only in an unsplit system, as explicitly recited in the claims. In contrast, the claims are directed to a split processing system, wherein there are a plurality of nodes, each node including one or more processors. The claims refer to split and unsplit systems so as to contrast the two types of systems and to provide an understanding for the limitations directed to greater availability.

In view of the above remarks, withdrawal of the § 112 rejection is respectfully requested.

¹ Definitions of a "node" are provided in page 11, lines 21-30 of the specification.

Prior Art Rejection

Claims 1, 13-15, 16 and 22-24 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Leymann et al. (hereafter, "Leymann"). Applicants respectfully traverse this rejection.

Applicants incorporate herein the arguments presented in the previous response. The remarks in section 2 below address the Examiner's new reasoning presented in the Final Rejection.

2. Patentability of claims 1 and 16 over Leymann

Claims 1 and 16 read as follows (underlining added for emphasis):

1. A split processing system comprising:

- (a) a plurality of nodes, each node including one or more processors
- (b) a communication network that allows the one or more processors at each of the nodes to cooperate with each other; and
- (c) means to allow one or more of the nodes to take over processing capacity of a node that becomes lost,
the availability of the split processing system being greater than the availability of an unsplit system wherein all of the processors are located at a single node.

16. A split processing system comprising:

- (a) a plurality of nodes, each node including:
 - (i) a processor subsystem including at least one processor, and
 - (ii) an operating system;
- (b) a communication network that allows the one or more processors at each of the nodes to cooperate with each other; and
- (c) means to allow one or more of the nodes to take over processing capacity of a node that becomes lost,
the availability of the split processing system being greater than the availability of an unsplit system wherein all of the processors are located at a single node.

a. Leymann does not disclose or suggest clause (c)

Regarding clause (c) in claims 1 and 16, as discussed in the previous response, Leymann cannot allow one of the nodes to take over processing capacity of a node that becomes lost because each node contains a unique portion of the database. Therefore, Leymann does not disclose or suggest clause (c).

Notwithstanding this fact, in the Final Rejection, the Examiner refers to four text portions of Leymann as allegedly disclosing this limitation. Applicants have carefully reviewed these four text portions and do not believe that any of the text portions disclose or suggest this limitation. The text portions are discussed below.

Text portion 1 (column 3, lines 40-58)

The introduction of parallel database technology within workflow management systems allows the latter to cope with the increasing requirements and expectations. Storing portions of this system repository in parallel databases leads to significant improvements with respect to parallelism, concurrency and availability. Parallel databases allow to work on part of the data at a time, cutting the time required for the operation to a manageable size. Partitioned tables allow a program to work on part of the data at a time, while allowing concurrent access to other programs on other partitions. It becomes possible to put more frequently accessed data on faster devices. More frequently accessed data can be separated from the remainder and can be put in a partition of its own and can use different device type. A single query to a partitioned database can initiate multiple parallel operations. These smaller queries run simultaneously on multiple processors accessing data in parallel. This reduces the elapsed time for a query.

This text portion in Leymann discusses how parallel databases improve availability.

(This text portion does not even state that partitioned tables (which is the invention in Leymann) accomplishes this goal.) There is no discussion in this text portion about nodes taking over from other nodes, or what happens if one of the databases at a node becomes lost. Accordingly, this text portion is not relevant to clause (c).

Text portion 2 (column 7, lines 15-33)

Connectors link activities in a process model. Using connectors, one defines the sequence of activities and the transmission of data between activities. Since activities might not be executed arbitrarily they are bound

together via control connectors. A control connector might be perceived as a directed edge between two activities; the activity at the connector's end point cannot start before the activity at the start point of the connector has finished (successfully). Control connectors model thus the potential flow of control within a business process model. Default connectors specify where control should flow when the transition condition of no other control connector leaving an activity evaluates to true. Default connectors enable the workflow model to cope with exceptional events. Data connectors specify the flow of data in a workflow model. A data connector originates from an activity or a block, and has an activity or a block as its target. One can specify that output data is to go to one target or to multiple targets. A target can have more than one incoming data connector.

This text portion is related to "data connectors." As described in column 7, lines 9-10 of Leymann, data connectors represent the transfer of data from output containers to input containers. This text portion appears to be completely unrelated to nodes becoming lost, or nodes taking over processing capacity from other nodes. Accordingly, this text portion is also not relevant to clause (c).

Text portion 3 (column 7, line 60 through column 8, line 2)

Process definition includes modeling of activities, control connectors between the activities, input/output container, and data connectors. A process is represented as a directed acyclic graph with the activities as nodes and the control/data connectors as the edges of the graph. The graph is manipulated via a built-in, event-driven, CUA compliant graphic editor. The data containers are specified as named data structures. These data structures themselves are specified via the DataStructureDefinition facility.

This text portion describes data structure details and also appears to be completely unrelated to nodes becoming lost, or nodes taking over processing capacity from other nodes. Accordingly, this text portion is also not relevant to clause (c).

Text portion 4 (column 10, line 63 through column 11, line 49)

In a parallel system, a database can be split into several separate parts, called partitions or sometimes called nodes. Each table in a database may be split into partitions. Each partition can run on a separate machine; it has its own log and its own set of indexes.

Two kinds of parallelism can be distinguished can be applied to the processing of an SQL statement. Intra-partition parallelism refers to simultaneous processes within a single partition, and inter-partition parallelism refers to simultaneous processes in multiple partitions. Both types of parallelism are orthogonal to each other, i.e. both techniques can be combined and applied in parallel.

Intra-partition parallelism is typically used on symmetric multiprocessor (SMP) machines, in which multiple processors share common memory and disks. Inter-partition parallelism is completely managed by the database management system and does not require any particular action on the user side except that the database management system must be told to exploit intra-partition parallelism.

Inter-partition parallelism works with a set of processors, each of which has its own memory and disks. Nothing is shared between the different processors. This hardware configuration is called a shared nothing system. Some or all of the processors itself may be symmetric multiprocessors that can then exploit intra-partition parallelism, thus intra-partition parallelism and inter-partition parallelism can be combined as orthogonal concepts. A typical machine that implements this architecture is the IBM RD6000 SP2. The different nodes processors communicate with each other via the clusters internal high speed network.

Each of the different partitions, which DB2 UDB calls a node, is assigned a unique identifier, in DB2 UDB an integer starting with 0. Each node is then assigned to the processor that hold the partition. The tables that are assigned to different partitions (called partitioned tables) must each have a partitioning key that determines how the rows of the tables are distributed among the partitions. The values of the partitioning key are then mapped to the set of partitions using a hash function. This hash function is supplied by the database management system. Such a situation is visualized in FIG. 2. A single logical table (200) actually is stored as a set of partitioned tables (201) an (202). Partitioning keys of the individual records are used to determine which record has to be stored in which partition table. In the example of FIG. 2 records with partitioning keys in the range of -1 to 10 (refer to 205 to 206) are stored in the partition of node 0 (201) while records with partitioning keys in the range of -71 to 99 (refer to 207 to 208) are stored in the partition of node 1 (202). Access to the logical table (200) results in inter-parallel access to the individual partitions (201) to (202).

A database client connects to one of the nodes. This node, the coordinator node, is responsible for all client requests and distributing these requests to the individual nodes.

This text portion describes certain details of Leymann's scheme for increasing interoperability of a workflow management system (WFMS) and a database management system (DBMS) wherein a database is split into a set of nodes with corresponding partitioned tables. Each node has a processor, and the node processors communicate with each other via a network. This text portion does not describe what happens if a node becomes lost, or whether any nodes take over processing capacity from lost nodes. Accordingly, this text portion is also not relevant to clause (c).

In sum, Leymann does not provide any disclosure or suggestion of clause (c).

b. Leymann does not disclose or suggest the availability limitations

Regarding the "availability" limitations in claims 1 and 16, as discussed in the previous response, Leymann does not disclose or suggest that there is greater availability in Leymann's split system compared to the original unsplit system. In fact, it is likely that there is less availability in Leymann as a result of splitting the database in the manner that Leymann performed the split (i.e., providing only unique portions of the database at each node). Leymann therefore provides no teaching of the increasing availability compared to an unsplit system. In fact, since it is likely that there is less availability in Leymann as a result of splitting the database, Leymann likely teaches against the increased availability limitations in claims 1 and 16.

In the paragraph numbered 12 on page 6 of the Final Rejection, the Examiner again refers to column 5, lines 5-9 of Leymann to support the availability limitation. However, as discussed on pages 11-12 of the previous response, this text portion of Leymann (referred to as the "fourth text portion" in the previous response) merely discusses how Leymann's invention maximizes parallelism, concurrency and availability, but does not state that availability is increased. This text portion merely states that given these three goals, Leymann's invention offers a maximized solution. This is not a clear disclosure of increasing availability. The Examiner further asserts that "the very nature of a parallel system as taught by Leymann et al. provides optimum availability." However, "optimum availability" is not necessarily "increased availability."

On page 5 of the Final Rejection, the Examiner admits that Leymann does not explicitly teach that availability of a split processing system is greater than the availability of an unsplit

system, but that Leymann's discussion that "performance impacts" can be minimized by system splitting provides a sufficient disclosure for an artisan to conclude that Leymann's system increases availability. Applicants respectfully disagree. There are many ways to minimize performance impacts that do not result in any increased availability. Some ways to minimize performance impacts even result in decreased availability.

In sum, Leymann also lacks any disclosure or suggestion of the claimed "availability" limitations.

For at least the reasons set forth above, withdrawal of the prior art rejection is respectfully requested.

3. Patentability of dependent claims 27-28

These claims were newly presented in the previous response and were not rejected over Leymann in the Final Rejection. These arguments for patentability are repeated from the previous response.

Leymann does not disclose or suggest a split processing system wherein each node has less processors than the number of processors in the unsplit system. Leymann is completely silent as to how many processors are present before and after the system splitting. Furthermore, these claims are believed to be allowable because they depend upon respective allowable independent claims.

4. Patentability of dependent claims 29 and 31

These claims were also newly presented in the previous response and were also not rejected over Leymann in the Final Rejection. These arguments for patentability are repeated from the previous response.

In the outstanding Office Action, the Examiner admits that Leymann does not explicitly teach the original claim limitations directed to reduced failure modes. However, the Examiner states that this limitation would have been obvious in view of Leymann's disclosure of improving performance. Stated simply, the Examiner presumes that as performance goes up, the number of failure modes (which is generally inversely related to availability) goes down. However, this is an incorrect assumption. Improving performance does not necessarily reduce

failure modes. In fact, the manner in which Leymann improves performance, namely, by splitting a database into unique portions, creating a plurality of nodes, and placing a unique portion in each node, will likely increase the number of failure modes. Leymann likely teaches away from claims 29 and 31. Furthermore, these claims are believed to be allowable because they depend upon respective allowable independent claims.

5. Patentability of dependent claims 30 and 32

These claims were also newly presented in the previous response and were also not rejected over Leymann in the Final Rejection. These arguments for patentability are repeated from the previous response.

Leymann has no disclosure of how to calculate failure modes or availability. Also, Leymann does not even discuss “spares.” Furthermore, these claims are believed to be allowable because they depend upon respective allowable independent claims.

6. Patentability of dependent claims 2-15, 17-26

The dependent claims are believed to be allowable because they depend upon respective allowable independent claims, and because they recite additional patentable steps.

Conclusion

Insofar as the Examiner’s rejections were fully addressed, the instant application is in condition for allowance. Withdrawal of the Final Rejection, entry of this response, and issuance of a Notice of Allowability of all pending claims is therefore earnestly solicited.

Respectfully submitted,

WILBUR H. HIGHLEYMAN et al.

August 7, 2006
(Date)

By: Clark Jablon
CLARK A. JABLON
Registration No. 35,039
AKIN GUMP STRAUSS HAUER & FELD LLP
One Commerce Square
2005 Market Street - Suite 2200
Philadelphia, PA 19103-7086
Telephone: (215) 965-1200
Direct Dial: (215) 965-1293
Facsimile: (215) 965-1210
E-Mail: cjablon@akingump.com

CAJ:gem